

## 第 15 课，写一个字节到 24c02 中

24c02 是一个非挥发 eeprom 存储器器件，采用的 IIC 总线技术。24c02 在许多试验中都有出现。24c02 的应用，主要在存储一些掉电后还要保存数据的场合，在上次运行时，保存的数据，在下一次运行时还能够调出。

24c02 采用的 IIC 总线，是一种 2 线总线，我们在试验中用 IO 来模拟这种总线，至于总线的时序和原理，请参考相关资料。如果您不想研究，也没有关系，我们在程序中已经为你写好了，现在和今后您都可以只调用就是，不必花时间和精力去研究。

一块 24c02 中有 256 个字节的存储空间。

我们将 24c02 的两条总线接在了 P26 和 P27 上，因此，必须先定义：

```
sbit SCL=P2^7;
```

```
sbit SDA=P2^6;
```

在这个试验中，我们写入了一个字节数值 0x88 到 24c02 的 0x02 的位置。

写入完成后，P10 灯会亮起，我们再在下一颗来读出这个字节来验证结果。

```
#define uchar unsigned char //定义一下方便使用
#define uint unsigned int
#define ulong unsigned long
#include <reg52.h> //包括一个 52 标准内核的头文件
//本课试验写入一个字节到 24c02 中
```

```
char code dx516[3] _at_ 0x003b;//这是为了仿真设置的
```

```
#define WriteDeviceAddress 0xa0 //定义器件在 IIC 总线中的地址
```

```
#define ReadDviceAddress 0xa1
```

```
sbit SCL=P2^7;
```

```
sbit SDA=P2^6;
```

```
sbit P10=P1^0;
```

```
//定时函数
```

```
void DelayMs(uint number)
```

```
{
    uchar temp;
    for(;number!=0;number--)
    {
        for(temp=112;temp!=0;temp--);
    }
}
```

```
//开始总线
```

```
void Start()
```

```
{
    SDA=1;
```

```

        SCL=1;
        SDA=0;
        SCL=0;
    }

//结束总线
void Stop()
{
    SCL=0;
    SDA=0;
    SCL=1;
    SDA=1;
}

//测试 ACK
bit TestAck()
{
    bit ErrorBit;
    SDA=1;
    SCL=1;
    ErrorBit=SDA;
    SCL=0;
    return(ErrorBit);
}

//写入 8 个 bit 到 24c02
Write8Bit(uchar input)
{
    uchar temp;
    for(temp=8;temp!=0;temp--)
    {
        SDA=(bit)(input&0x80);
        SCL=1;
        SCL=0;
        input=input<<1;
    }
}

//写入一个字节到 24c02 中
void Write24c02(uchar ch,uchar address)
{
    Start();
    Write8Bit(WriteDeviceAddress);
    TestAck();
}

```

```
Write8Bit(address);
TestAck();

Write8Bit(ch);
TestAck();

Stop();
DelayMs(10);
}

//本课试验写入一个字节到 24c02 中
void main(void) // 主程序
{
    Write24c02(0x88,0x02);// 将 0x88 写入到 24c02 的第 2 个地址空间

    P10=0; //指示运行完毕
    while(1); //程序挂起
}
```

编译，联机进入仿真，等待 LED 亮起。本课结束。我们在下一课看结果。

作页：无